Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5580--05-8884

A System with Intelligent Editing for Extracting Ridge and Ravine Terrain Features

GREG SCHMIDT

ITT Industries
Advanced Information Technology Branch
Information Technology Division

J. EDWARD SWAN, II
LAWRENCE ROSENBLUM
ERIK B. TOMLIN

Advanced Information Technology Branch
Information Technology Division

DEREK OVERBY

Graphics Software Developer
Computing and Information Technology Division
Texas Center for Applied Technology

June 30, 2005

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
30-06-2005	Memorandum Report	June 2001-June 2002

4. TITLE AND SUBTITLE

A System with Intelligent Editing for Extracting Ridge and Ravine Terrain Features

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N0001404WX20053

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Greg S. Schmidt,* J. Edward Swan, II, Lawrence Rosenblum,
Erik B. Tomlin, and Derek Overby†

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory, Code 5580
4555 Overlook Avenue, SW
Washington, DC 20375-5320

ITT Industries

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/5580--05-8884

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)
AIT-05-032

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

*ITT Industries, Naval Research Laboratory, Code 5580
†Graphics Software Developer, Computing and Information Technology Division, Texas Center for Applied Technology, 214 Wisenbaker Engineering Research Center

14. ABSTRACT

We describe a system for extracting ridges and ravines from elevation data. The application context is a map-based military planning tool, which allows users to select ridges and ravines by simple mouse clicks. The extracted terrain features are complete in the application-specific sense that they conform to what our users expect a single ridge or ravine to look like. Supervision is supported by a graphical user interface, which allows an analyst to modify algorithm parameters as well as perform intelligent mouse-based editing operations. Among similar existing systems, ours is unique in that it focuses on the three classically difficult operations of (1) combining partial features, (2) splitting multiply-connected features, and (3) removing micro-terrain features for extracting high-level structure from classical pixels.

15. SUBJECT TERMS

Feature extraction; Terrain visualization; Image segmentation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Greg S. Schmidt
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UL	18	**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified			(202) 767-0371

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

CONTENTS

A SYSTEM WITH INTELLIGENT EDITING FOR EXTRACTING RIDGE AND RAVINE TERRAIN FEATURES

INTRODUCTION

Technologies for collecting digital terrain data, such as satellite imaging, continue to advance at a rapid pace, which is making an ever-increasing amount of terrain data available for a wide variety of applications. Many of these applications require specific features of the data, such as rivers, ravines, forested areas, etc., to be extracted and labeled. Developing an application-specific feature extraction system is difficult, and these systems usually require supervision by a highly skilled analyst. This motivates the considerable amount of work that has been performed in the terrain feature extraction field.

Although the ultimate goal of this field remains completely automatic, unsupervised extraction, typically success involves minimizing the amount of supervision required. The work presented in this paper falls into this category: we describe a system for extracting ridges and ravines from elevation data, where the extracted terrain features are *complete* in the application-specific sense that they conform to what our users expect a single ridge or ravine to look like. Our application area is a system that allows military planning operations using multimodal combinations of voice commands and pen-based gestures[5]. The extraction system allows us to implement commands such as "place a fortification along **this ridge**", or "make a new mechanized battalion in **this ravine**", where the phrases in italics indicate the simultaneous selection of a terrain feature by a pen-based gesture.

Existing terrain feature extraction efforts can be classified into two major groups: systems which extract features at a local pixel level (e.g., Haralick[15], Peucker and Douglas[32]), and systems which extract high-level structure from classified pixels or other lower-level structures (e.g., Mark[24], Kass[18]). To date, neither approach has been able to completely extract terrain features. Our system fits into the second group; we utilize the well-known Topographic Primal Sketch technique of Haralick, et al.[15] to perform pixel-level classification. We then extract high-level structure from the classified pixels by focusing on three classically difficult operations: (1) combining partial features, (2) splitting multiply-connected features, and (3) removing micro-terrain features. These operations are illustrated in Figure 1. As described below, this focus has given good results for extracting ridges and ravines from elevation data, with relatively minimal supervision. We believe we are among the first groups to apply this particular focus to the terrain feature extraction domain.

Manuscript approved April 21, 2005

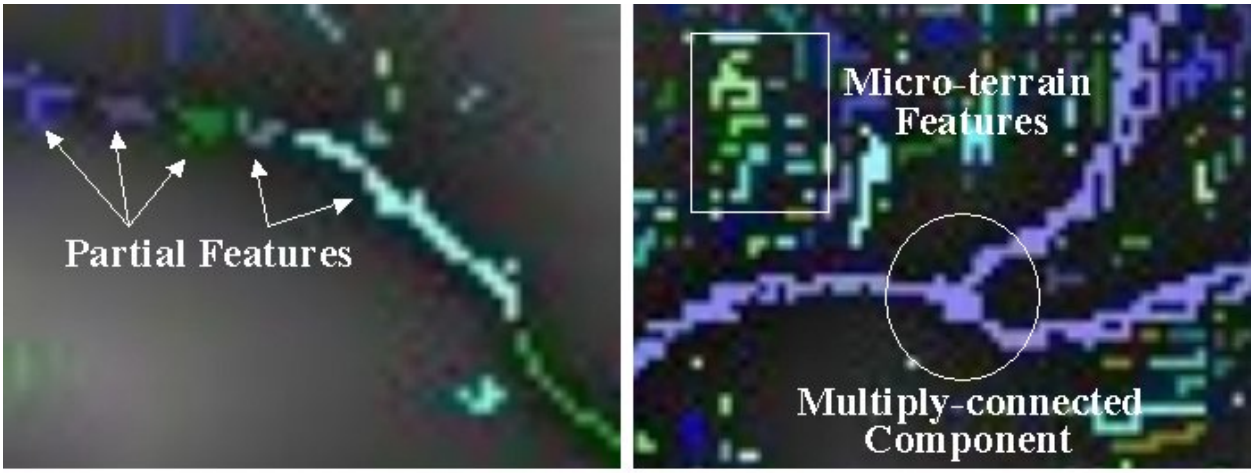

Fig. 1 — Difficult Feature Extraction Operations

In Related Work we discuss related work in the areas of local pixel-level extraction and structure extraction schemes. In System Design we describe our system's architecture, user interface, and give results. We follow in Conclusions and Future Work with conclusions, a discussion of our system's contributions, and our plans for future work.

RELATED WORK

This section lists relevant work from the feature extraction literature. More complete details can be found in a survey by Mascardi[25].

Local Extraction Schemes

Local extraction schemes operate by matching local topological traits with pre-determined patterns. Our system is based on the popular Topographic Primal Sketch algorithm by Haralick et al.[15]. This algorithm classifies terrain pixels into basic topological types (ravines, ridges, peaks, etc.) by using patterns formed from surface gradients. Similar schemes have been designed by Peucker and Douglas[32], Toriwaki and Fukumura[39], Fowler and Little[10], Skidmore[35], and Gauch and Pizer[11]. While all of these methods effectively extract topological features at the pixel level, they do not yield any information about which particular groups of pixels form complete features.

Structure Extraction Schemes

A popular technique for extracting structure for ravines is hydrological analysis, which uses slopes of maximum descent to form a drainage network (see Jenson and Domingue[17], Mark[24], and Band[1]). These methods apply principles of water flow to connect ravines. To a degree, ridges can be similarly computed by inverting the data. However, in flat areas these methods produce many false ravine and ridge components.

Another popular method, Snakes[18], is used to determine curvilinear structures in topographical data. The essential idea is to stretch and bend a topographically-based spline until it fits the structure of curvilinear terrain features. This method requires a lot of effort in initialization, the underlying spline representations, and the formulation of the stretching and bending operations. Steger[36] proposes a similar method which utilizes differential geometric properties of the terrain data. Thompson et al.[38]

have demonstrated good results in extracting micro-terrain ravines and their boundaries by integrating the hydrological analysis and snakes techniques.

Related techniques have been applied for extracting road networks (Guindon[13], Tupin et al.[40]), as well as building and tree boundaries (Gerke et al.[12]). Ravines and ridges are easily confused with roads and tracks.

From the point of view of our application domain, the primary drawback of all of these techniques is that they do not give any structural information that can be used to separate multiply-connected features. For example, in the situation where multiple ridges come together, our application requires us to be able to refer to each ridge in isolation.

NIMA Products

The National Imagery and Mapping Agency (NIMA) has developed a standard for storing terrain feature information called Vector Product Format[30]. They have databases of features for topography, littoral, hydrographic and aeronautical data. Some of the features for the topographic data include boundaries, elevation and ground obstacles. The primary drawback is that NIMA's systems and algorithms are proprietary, and thus cannot be adapted to specific circumstances.

SYSTEM DESIGN

Our system takes digital terrain elevation data and extracts complete ridges and ravines in vector format. The system is composed of a data processing pipeline with eight major steps. Figure 2 displays a flow diagram for these steps. Each step is indicated by a shaded box, where the shade (dark gray, light gray, white) signifies the degree of operator supervision required (substantial, medium, minimal). Medium supervision takes the form of several adjustable parameters that are accessed using our graphical user interface (GUI) (see User Interface and Interaction and Figure 7). Substantial supervision involves a combination of intelligent mouse-based image editing and parameter setting. Data flows in Figure 2 are labeled (a) through (h). Figure 3 gives an image from each data flow for an example dataset.

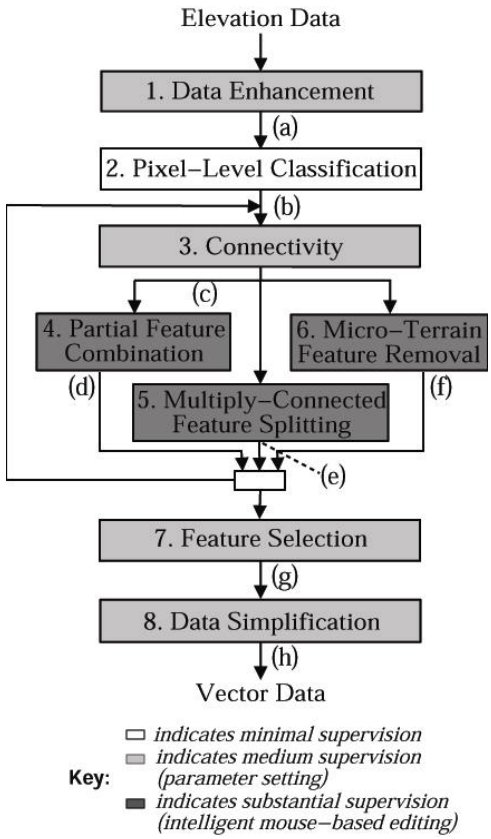

Fig. 2 — Segmentation Processing Pipeline. Figure 3 Shows Data Flows Labeled (a) through (h).

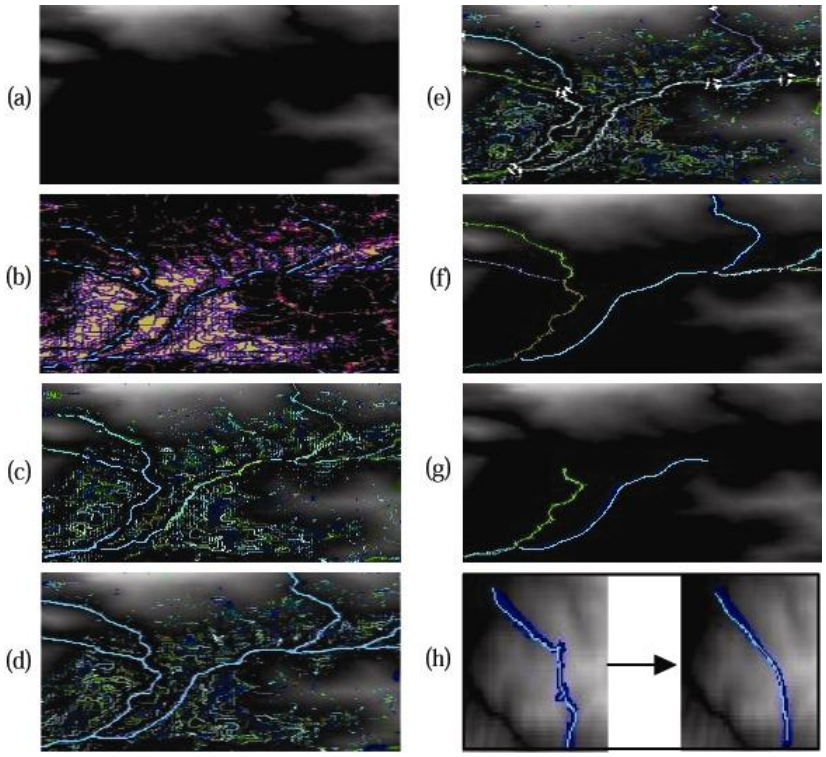

Fig. 3 — Images from the Labeled Data Flows in Figure 2.

Processing Pipeline

1. Data Enhancement:

The first step in the processing pipeline addresses resolution and contrast between neighboring pixels, since both of these properties make feature extraction easier. Higher resolutions yield better estimations for topological discriminants such as gradients, which require sub-pixel computations. Similarly, edge detection algorithms perform better as contrast increases.

In our system, we enhance contrast with a sharpening filter and an image intensity dynamic range modification method. For each pixel, the sharpening algorithm[22] creates a high-pass filter from a local region histogram, which filters out low-frequency pixel intensities. The dynamic range modification method[33] modifies the global histogram by increasing the separation of the pixel intensity classes. The application of this algorithm results in small height adjustments that increase local contrast. Medium supervision is required for these operations.

2. Pixel-Level Classification:

In the second processing step we locally classify data points by assigning attributes of terrain features at the pixel level. We utilize the well-established Topographic Primal Sketch algorithm[15], which classifies by matching predetermined topological patterns built on first- and second-order surface gradients. We estimated these gradients by fitting cubic splines to the elevation data. Similar methods have been designed by Peucker and Douglas[32], Toriwaki and Fukumura[39], Fowler and Little[10] and Skidmore[35]. They all generate a set of basic terrain feature types, which serve as the building blocks for forming complete terrain features. The accuracy of these algorithms improves as the data resolution increases. Minimal supervision is typically required for this step.

3. Connectivity:

The limitation of the local terrain pixel classifications is that they do not provide any high-level structural information that describes which sets of terrain pixels form complete features. This motivates the current step, which is to cluster like-classified data points together. Most data clustering problems arise from sampling resolution issues. Data that is sampled too coarsely produces “holes”, which greatly affects the performance of the connectivity algorithms. In this case, the connectivity algorithms usually generate sets of partial features. High resolution sampling improves the performance of the connectivity algorithms, but does not completely eliminate problems. By increasing the resolution, more fine-detail data is added to the datasets. While this allows the algorithms to better connect the components of large-scale features, high resolution can also cause *micro-terrain* features, which we define as features that are one or more resolution scale-factors smaller than the current resolution.

Another problem with connectivity algorithms occurs when multiple terrain features connect to each other, such as two rivers. It is not an easy problem to automatically separate the two features. It appears that more sophisticated algorithms or additional knowledge are required to deal with these types of problems. This also applies to separating micro-terrain features, noise and those features that are unimportant for the current task.

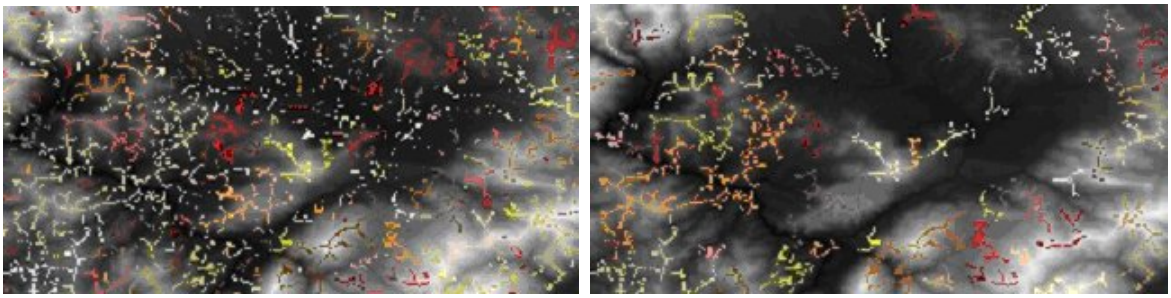

Fig. 4 — (Left) This image shows a large number of small micro-terrain features. (Right) Some of the clutter is reduced by removing the smallest 5% of the features.

Our system implements a connectivity algorithm that connects the current pixel with like-classified neighboring pixels up to *k* pixels away. For *k = 1*, the algorithm connects the current pixel's immediate neighbors if and only if they have the same classification. With values of *k > 1*, the neighbors less than *k* distance away can have any classifications. The algorithm creates *k*-connected components, which consist of either partial, whole, multiply-connected, or micro-terrain features. It is rare to produce a whole feature. Different values for *k* affect how many partial or multiply-connected features are produced. Small values produce more partial features, while large values produce multiply-connected features.

The software interface has features that allow the parameter *k* to be adjusted interactively by the user. The user can also choose to remove sets of small and large clusters interactively. These interface options provide some control for clustering the data into connected components; however, they still leave partial, multiply-connected, micro-terrain features, noise and unimportant features. This motivated the development of the next three steps in the architecture. A medium level of supervision is required.

4. Partial Feature Combination:

This processing step is focused on combining the partial features to make whole terrain features. The *combining operation* is very difficult to automate by computer since it is hard to determine which components make up one feature. Typically ridges and ravines are connected in complicated curvilinear structures that may bifurcate into many sub-branches. Methods that can be applied for the *combining operation* are hydrological analysis and Snakes, as discussed above in Related Work (Related Work). Recall that each of these methods has drawbacks related to their effectiveness and accuracy.

We implemented an intelligent mouse-based image editing method that combines drawing and erasing operations with parameter setting operations. The software interface provides controls for selecting drawing modes corresponding with individual feature types and offers options to modify parameter settings for the connectivity, cluster thresholding and other algorithms. Once the appropriate drawing mode is selected, the interface works similar to a standard mouse-driven drawing package. A nice feature of our implementation is the capability to see immediately the effect that each drawing or erasing operation has on the high-level structure of the locally classified pixels. This operation is performed by re-executing the connectivity, cluster thresholding and related algorithms after each drawing operation. Figure 2 indicates this iteration by the backwards flowing arrow that leads from before *step 7* to before *step 3*. Unfortunately this step requires significant supervision.

5. Multiply-Connected Feature Splitting:

Splitting multiply-connected features is a difficult operation to perform. Candidate *splitting operations* are boundary tracing algorithms (e.g., Crawford-Hines et al.[7] and Burton[4]) and Snakes

(described previously). Most boundary tracing algorithms have been shown to work for only specific datasets and problem domains. The Snakes method requires a lot of supervised overhead and has other drawbacks. The most effective *splitting operations* for our domain that we are aware of are the intelligent mouse-based image editing operations we implemented in *step 4*.

6. Micro-Terrain Feature Removal:

The micro-terrain features and noise are removed in this processing step. The micro-terrain features have a scale that is a magnitude smaller than the current sampling interval. Because of this, the connectivity algorithms typically produce many small clusters. Noise is typically in the form of small clusters as shown in Figure 4 (Left). These features can be removed by eliminating the smallest size connected components (Figure 4 (Right)).

For the remaining micro-terrain features, some are large connected components and the rest are attached to normal terrain features. The *splitting operation* used in *step 4* can be applied to the attached micro-terrain features. However, determining how to remove the remaining micro-terrain features is more difficult. These features typically appear to have more complex or compact shapes. This knowledge can be applied to develop an automatic removal algorithm; however, in our system, we simply select them by highlighting them manually and remove them.

7. Feature Selection:

We have found that terrain segmentation can be made very powerful with the capability to select features based on their geometrical characteristics such as length, slope and elevation. Additionally, generating ranked lists of features based on these characteristics adds to that power. Operations based on these can yield feature sets such as: “the segments of a river containing the lowest elevation points” and “the ridges containing the highest peaks”. This capability can be even further enhanced by applying Boolean operators to the selections. We implemented these selection capabilities into our system.

Our system can handle many types of terrain feature characteristics such as: the longest, those containing the highest peaks or lowest pits, those with the largest or smallest change in elevation, those with the highest or lowest average elevation, and those containing the largest or smallest average gradients. We also integrated a Boolean-based selection pipeline, so that new combinations of algorithms can be made to operate on the remaining selections. The pipeline works by applying a ranked list of operations to the current feature selections. For example, the user can rank “the 10 longest features” as “1” and “the 5 features containing the highest peaks” as “2”. Then the pipeline will apply the algorithms in their ranked order to the current selected components. The operations can also be run in parallel and combined as a Boolean “union” if desired as well. Additionally, the algorithms can be applied separately or in combination to specific raw feature types such as ridges and ravines. These algorithms require minimal supervision and operate very quickly.

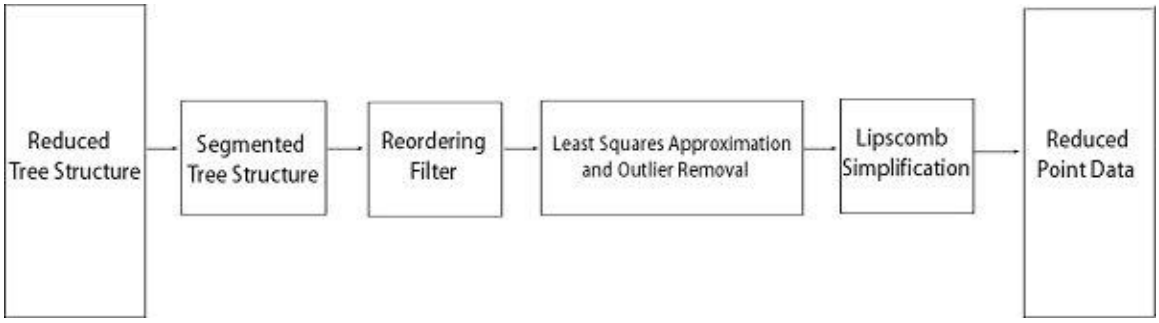

Fig. 5 — Simplification Pipeline for Removing Redundant Elements in the Selected Features

8. Data Simplification:

The last processing step is to simplify the data such that redundant data points are removed. This step produces data that can be used to quickly and accurately re-create the terrain features while minimizing storage space requirements. Some of the common sources of data redundancy that can be eliminated are: (1) thick multi-pixel boundaries, (2) points lying outside the general shape of the terrain features, and (3) redundant structural components which may be points, lines or polygonal objects.

For example, when ravines and ridges are extracted, they typically contain multi-pixel thick connected components. These features can be ``thinned'' by using Rosenfeld and Kak's thinning algorithm[34]. Ravines and ridges typically contain complex twists and turns. If desired, some of these can be smoothed out by fitting piecewise curves[9] to the features and removing outlier points[6]. Additionally, the smooth sections can be resampled or redundant points can be removed. For example, a linear component that contains more than two points can be reduced to the two end points.

For our system, we implemented a series of algorithms that remove redundancies in the data. Figure 5 shows the simplification pipeline we used to remove redundant elements in the selected features. We first segmented the tree structure data into branches and subsequently applied a curve reordering filter on the branches. We used a least squares curve fitting approach to remove thickness and outlier points[6]. Then we applied a cross product filter, which removes points if the angle a point makes with the previous and next point is less than some specified threshold[23]. These algorithms require minimal supervision. The three major steps of the simplification pipeline are illustrated in Figure 6.

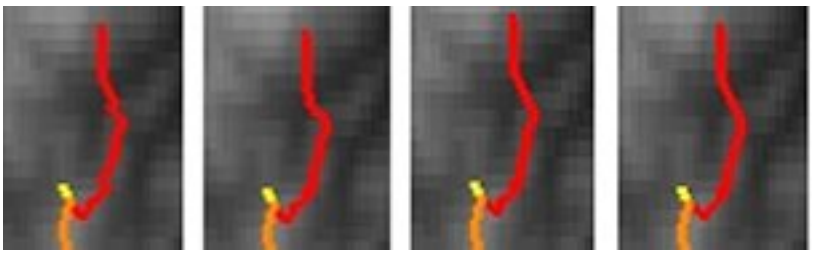

Fig. 6 — The three major steps of the simplification pipeline are illustrated here. From left to right, the curve data is (a) segmented and re-ordered, (b) a least squares fitting is performed upon the segments and (c) the segments are smoothed using a cross product filter.

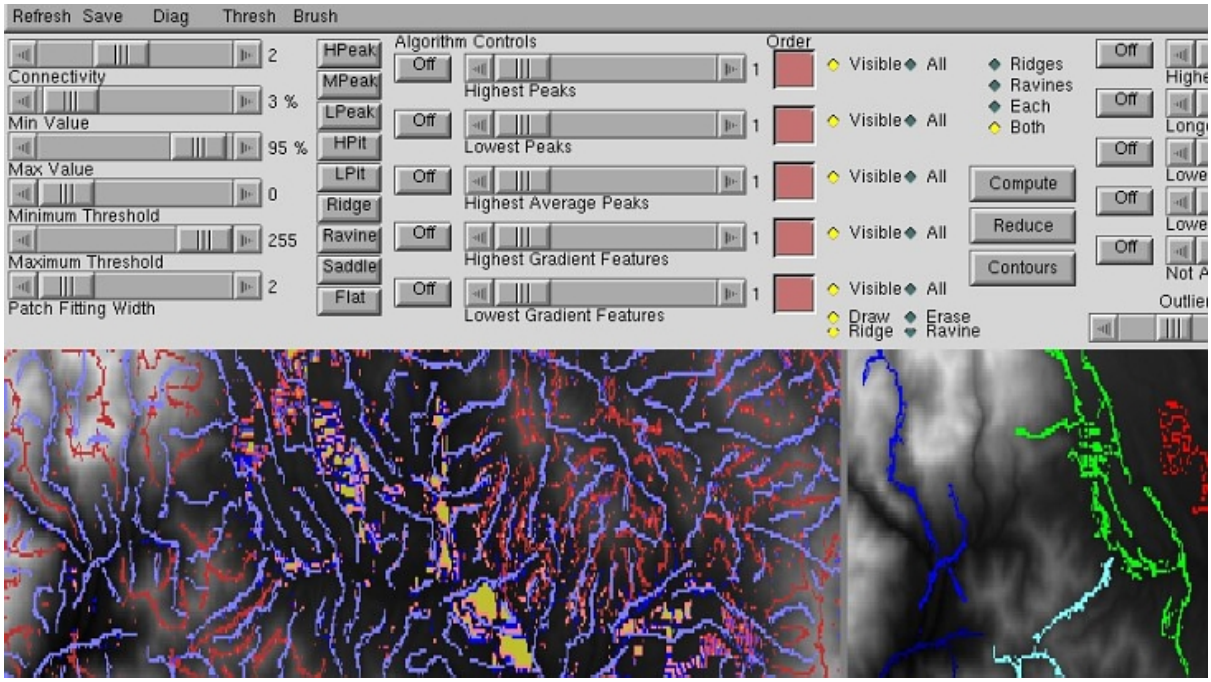

Fig. 7 — A Portion of the Graphical User Interface of the System

The final output from our system is the extracted terrain features stored in a vector format.

User Interface and Interaction

Figure 7 shows our system's graphical user interface (GUI), which provides controls for the required supervision. This supervision is in two forms: parameter setting, and intelligent mouse-based image editing operations. Our GUI contains two display windows and multiple interaction controls including sliders, buttons, and text input windows. The two display windows complement each other, such that when a processing step is applied, one shows the input, and the other shows the output.

The GUI contains parameter-setting sliders for the histogram normalization, the sharpening filter, the *k*-connectivity algorithm, the windowed-thresholding scheme for removing percentages of the smallest and largest clusters, the feature selections, and the data simplification step. There are nine different sliders for feature selection and two for data simplification.

We designed buttons to allow the user to select or unselect the basic feature types from the pixel classification algorithm. The processing algorithms only work on the selected feature types. We also implemented buttons to connect (or trace) either individual or combinations of ridge-like or ravine-like elements. We combined peaks with ridges and pits with ravines when performing the *k*-connectivity algorithm, which is logical because the ridges are composed of peaks and ravines are composed of pits. The interface has buttons and text windows that allow the analyst to specify a ranking order and Boolean combinations of feature selection operations.

Walk-Through of the Pipeline

We demonstrate the pipeline by proceeding with a walk-through of each of the significant processing steps required to extract a set of features. The walk-through is shown in Figure 3. Image (a) shows a gray-scale sub-region of the 3D terrain elevation dataset after enhancement. Image (b) shows the raw pixel-

level terrain feature classifications (ridge, ravine, flat, etc.). Image (c) shows only the ravine elements, which have been clustered using the *k*-connectivity algorithm with *k=1*. Image (d) shows the significant features highlighted after the *combining operation*, (e) shows them with end-point indicators after the *splitting operation*, and (f) shows them after the micro-terrain features have been removed. Image (g) shows three ravines containing the lowest peaks, and (h) shows a portion of one segment being simplified.

DEMONSTRATION OF THE SYSTEM

We demonstrate our system by selecting a desired feature dataset. The features we selected are ridges and ravines that are significant and are non-bifurcating. By *significant*, we mean the largest segments that are not (and do not contain adjoining) micro-terrain features and noise data. Our choice of features is shown in Figure 8 (Left). The red curves are the ravines and the blue are the ridges. Our dataset did not include every significant ridge and ravine in the region, but most of them. The image also shows a terrain texture of the region to help identify what the area's geography looks like.

The dataset was generated by using the interface tools to select low-level pixel classifications, connect the data, edit the connectivity, remove undesirable branches, select specific mathematical features and simplify the data. The desired features are output in a 3D vector format, which is displayed in Figure 8 (Right). These features were then overlaid on the 3D elevation terrain data to verify the registration. Figure 9 shows two images of the vector-formatted features overlaying the terrain raised up a few meters in height to make them more visible. In some cases the features appear to have poor registration (see the bottom of Figure 9 (Left)). However, this is not the case. The apparent mis-alignments result from the interactive curve simplification process. The degree of simplification is adjusted by the user and can be used to eliminate certain topological features in the ridge and ravines selections if desired. For the case above, the degree of simplification was set too high.

The red and blue features on the terrain texture map were added after this step and utilized to make this step more understandable for the comparison. Initially we compared the vector-formatted features with the texture in Figure 8 (Left) and the 3D elevation data. To make the texture maps we merged and enhanced the 2D orthogonal projection of the 3D vector data with a colored texture image. We show two portions of the dataset in two different visualization packages in Figure 10. The left image shows a hilly area with the ravines (red) and ridges (blue) highlighted using a software package called 3D Terrain Visualizer (3DTV)[19]. The right image shows a mountainous region in the software Dragon[8] with the main ridge highlighted in beige and other minor features showing up in different colors.

CONCLUSIONS AND FUTURE WORK

We have described a system for extracting ridges and ravines from elevation data. Our system starts with a pixel-level classification scheme, and then extracts high-level structures by iteratively performing the following operations: (1) combining partial features, (2) splitting multiply-connected features, and (3) removing micro-terrain features. The supervision required by these steps is provided by a GUI that allows graphical parameter setting as well as intelligent mouse-based image editing.

There are three aspects of our system which we believe make it a contribution to the terrain feature extraction field. First, while acknowledging that there are many proprietary algorithms (e.g., at NIMA[30] and similar places), we are not aware of other published work that extracts high-level structure by focusing on the three difficult operations of (1) combining partial features, (2) splitting multiply-connected features, and (3) removing micro-terrain features. Second, while applying our system to

additional linear terrain feature types may require additional operations, we have identified three operations that are fundamental in the sense that they will need to be addressed by any system that extracts complete high-level terrain structures from classified pixels. Finally, we are focused on extracting terrain features that are complete in the application-specific sense that they look like a "single" ridge or ravine. This focus gives us a point of uniqueness from most previous work.

In the future, we plan to further reduce the supervision required by our system. Each of our three operations is difficult, and as our literature review has demonstrated, each has already attracted considerable previous work. In the literature we have identified a number of techniques that should further reduce the supervision each operation requires. We also plan to extend our system to extract additional linear terrain features such as roads, rivers, agricultural boundaries created by field placement and crop type, urban boundaries, and building boundaries. This extension will test the generality of our system design.

Acknowledgements

We acknowledge Rob King, Augustine Su, Behzad Kamgar-Parsi, Pradeep Mistry, and Aaron Bryden for their contributions to the software and technical advice. This work was supported by the Office of Naval Research.

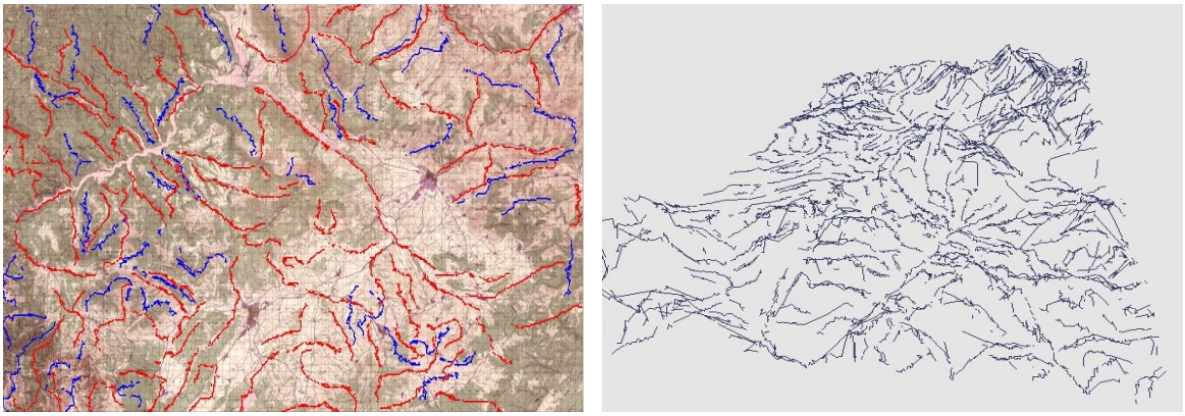

Fig. 8 — (Left) This image shows the dataset we selected which contains a majority of the most significant ridge (blue) and ravine (red) terrain features. (Right) The features extracted from our dataset are stored in 3D vector format and shown in 3D.

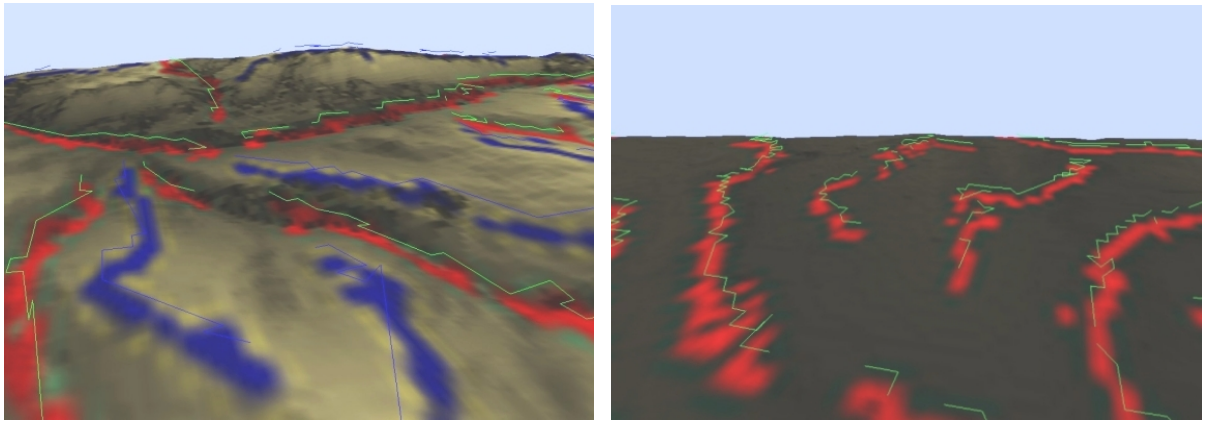

Fig. 9 — Both images show the features overlaid on the terrain elevation data. The left image also shows a ridge (blue) near the bottom in the middle that was over-simplified.

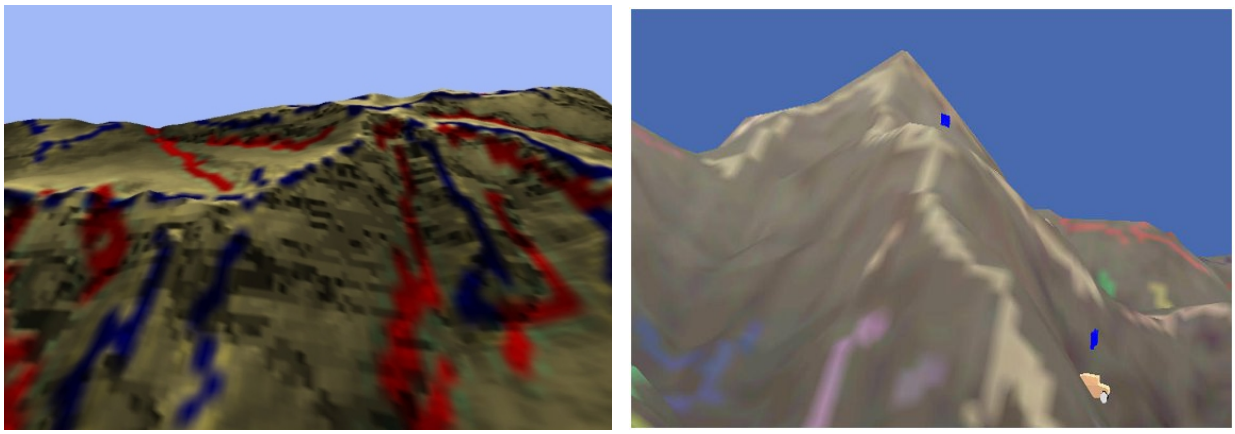

Fig. 10 — The images show the vector-formatted features combined and enhanced with the terrain texture. The left image shows a hilly area using the 3DTV[19] software with the features highlighted in red and blue (same as before). The right image shows a large ridge highlighted in beige in a mountainous region using the software Dragon[8].

REFERENCES

1. L. E. Band, "Topographic Partition of Watersheds with Digital Elevation Models," *Water Resources Research 1986*, vol. 22[15].

2. M. O. Berger and R. Mohr, "Towards Autonomy in Active Contour Models," *10th International Conference on Pattern Recognition 1990*, pp. 847-851.

3. S. Brunett and T. Gottschalk, "A Large-Scale Metacomputing Framework for the ModSAF Real-Time Simulation," *Parallel Computing 1998*, vol. 24, pp. 1873-1900.

4. B. P. Burton, *Chapter VIII: Segment Tracing, Automated 3D Reconstruction of Neuronal Structures from Serial Sections*, Texas A&M University, 1999.

5. P. Cohen and D. McGee and S. Oviatt and L. Wu and J. Clow and R. King and S. Julier and L. Rosenblum, "Multimodal Interaction for 2D and 3D Environments," *Projects In VR, IEEE Computer Graphics and Applications 1999*, vol. 19[4], pp. 10-13.

6. T. H. Cormen and C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, MIT Press, 1994.

7. S. Crawford-Hines and C. W. Anderson, "Neural Nets in Boundary Tracing Tasks," *IEEE Workshop on Neural Networks and Signal Processing 1998*, pp. 207-215.

8. J. Durbin and J. E. Swann II and B. Colbert and J. Crowe and R. King and T. King and C. Scannell and Z. Wartell and T. Welsh, "Battlefield Visualization an the Responsive Workbench," *IEEE Visualization 1998*, pp. 463-466.

9. G. E. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, 3rd Edition*, Academic Press, Inc., 1993.

10. R. J. Fowler and J. J. Little, "Automatic Extraction of Irregular Network Digital Terrain Models," *Computer Graphics 1979*, vol. 13, pp. 199-207.

11. J. M. Gauch and S. M. Pizer, "Multiresolution Analysis of Ridges and Valleys in Grey-Scale Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence June 1993*, pp. 635-646.

12. M. Gerke and C. Heipke and B. M. Straub, "Building Extraction from Aerial Imagery Using a Generic Scene Model and Invariant Geometric Moments," *Proceedings of the IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas 2001*, pp. 85-89.

13. B. Guindon, "Application of Spatial Reasoning Methods to the Extraction of Roads from High Resolution Satellite Imagery," *Geoscience and Remote Sensing Symposium Proceedings, IEE IGARSS 1998*, pp. 1076-1078.

14. N. Haala and V. Walter, "Automatic Classification of Urban Environments for Database Revision Using LIDAR and Color Aerial Imagery," *International Archives of Photogrammetry and Remote Sensing June 1999*, vol. 32.

15. R. M. Haralick and T. J. Laffey and L. T. Watson, "The Topographic Primal Sketch," *Journal of Robotics Research 1983*, vol. 2[1], pp. 50-72.

16. N. R. Harvey and S. P. Brumby and S. Perkins and J. Theiler and J. J. Szymanski and J. J. Bloch and R. B. Porter and M. Galassi and A. C. Young, "Image Feature Extraction: GENIE vs. Conventional Supervised Classification Techniques," *IEEE Transactions an Geoscience and Remote Sensing 2001*, pp. 100-111.

17. S. K. Jenson and J. O. Domingue, "Extracting Topographic Structure from Digital Elevation Data for Geographic Information Systems Analysis," *Photogrammetric Engineering and Remote Sensing 1988*, vol. 54[11], pp. 1593-1600.

18. M. Kass and A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision 1993*, pp. 321-331.

19. C. Kocmoud and J. Wall and G. Schmidt and D. Overby, *3D Terrain Visualizer (3DTV)*, Texas Center for Applied Technology, TEES, Texas A&M University, July 2000.

20. S. Lacroix and S. Fleury and H. Haddad and M. Khatib and F. Ingrand and G. Bauzil and M. Herrb and C. Lemaire and R. Chatila, "Reactive Navigation in Outdoor Environments," *International Journal of Robotics Research, Special Issue on Field and Service Robotics*, 1998.

21. K. Lai and R. Chin, "On Regularization, Formulation and Initialization of the Active Contour Models (snakes)," *Second Asian Conference on Computer Vision 1993*, pp. 542-545.

22. J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall PTR, 1990, pp. 606-609.

23. J. S. Lipscomb, "A Trainable Gesture Recognizer," *Pattern Recognition 1991*, vol. 24[9], pp. 895-907.

24. D. M. Mark, "Automated Detection of Drainage Networks from Digital Elevation Models," *Auto-Carto VI: Proceedings Sixth International Symposium on Computer Assisted Cartography 1983*, pp. 288-298.

25. V. Mascardi, "Extraction of Significant Terrain Features from RSG and TIN: A Survey," http://citeseer.nj.nec.com/mascardi98extraction.html, accessed March 1, 2003.

26. S. Menet and P. Saint-Marc and G. Medioni, "Active Contour Models: Overview, Implementation, and Applications," *IEEE International Conference on Systems, Man and Cybernetics 1990*, pp. 194-199.

27. R. Meth and R. Chellappa, "Target Indexing in Synthetic Aperture Radar Imagery Using Topographic Features," *Acoustics, Speech and Signal Processing, ICASSP-96 Conference Proceedings 1996*, pp. 2152-2155.

28. W. Neuenschwander and P. Fua and G. Szekely and O. Kubler, "Initializing Snakes," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 1994*, pp. 658-663.

29. M. Niklova and A. Hero, "Segmentation of Road Edges from a Vehicle-Mounted Imaging Radar," *9th IEEE SP Workshop on Statistical Signal and Array Processing Proceeding 1998*, pp. 212-215.

30. National Imagery and Mapping Agency, http://www.nima.mil, accessed January 14, 2002.

31. "RITN and ModSAF," http://ait.nrl.navy.mil/modsaf, accessed January 5, 2002.

32. T. K. Peucker and D. H. Douglas, "Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data," *Computer Graphics and Image Processing 1975*, vol. 4, pp. 375-387.

33. W. H. Press and S. A. Teukolsky and W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing, 2nd Ed.*, Cambridge University Press, 1992.

34. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, 1982.

35. A. K. Skidmore, "Terrain Position as Mapped from Gridded Digital Elevation Model," *International Journal of Geographical Information Systems 1990*, vol. 4[1], pp. 33-49.

36. C. Steger, "Extracting Curvilinear Structures: A differential Geometric Approach," *4th European Conference on Computer Vision (ECCV '96) Proceedings April 1996*, vol. 1.

37. G. W. Thoenen and W. B. Thompson, "Extraction of Micro-Terrain Ravines Using Image Understanding Constrained by Topographic Context," http://citeseer.nj.nec.com/197186.html, accessed 1996.

38. W. B. Thompson and G. W. Thoenen and R. G. Moore and T. C. Henderson, "Extraction of Micro-Terrain Features," *IMAGE Conference 1998*.

39. J. Toriwaki and T. Fukumura, "Extraction of Structural Information from Grey Images," *Computer Graphics and Image Processing 1978*, vol. 7[1], pp. 30-51.

40. F. Tupin and H. Maitre and J. F. Mangin and J. M. Nocolas and E. Perchersky, "Detection of Linear Features in SAR Images: Application to Road Network Extraction," *IEEE Transactions on Geoscience and Remote Sensing 1998*, vol. 36[2], pp. 434-453.

41. M. Wand and J. Evans and L. Hassebrook and C. Knapp, "A Multistage, Optimal Active Contour Model," *IEEE Transactions on Image Processing 1996*, vol. 5[11], pp. 1586-1591.

42. M. Woo and J. Neider and T. Davis, *OpenGL Programming Guide, 2nd Edition*, Addison-Wesley Developers Press, 1996.